I hereby certify that this correspondence is being deposited with the U.S. Postal Service as Express Mail, Airbill No. EV 296 583 748 US, in an envelope addressed to: MS Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below.

Dated: April 21, 2004 Signature: _____
(Margo Barbarash)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR U.S. LETTERS PATENT

Title:

BUSINESS RULES PREPROCESSING

Inventor:

Sekar Govindasamy

Ashley N. Moore
JENKENS & GILCHRIST, A
PROFESSIONAL CORPORATION
1445 Ross Avenue, Suite 3200
Dallas, Texas 75202
(214) 965-7393

BUSINESS RULES PREPROCESSING

BACKGROUND OF THE INVENTION

Technical Field

[0001] The present invention generally relates to multi-tiered architectures, and more particularly, but not by way of limitation, to a method of and system for transmitting transformed business rules from a business rules tier to a database tier for evaluation.

History of the Related Art

[0002] Multi-tiered infrastructures are implemented to provide a physical division of the infrastructure across, for example, desktop, local server, and central server processing nodes. As shown in Figure 1, in a multi-tier architecture 100, there are three or more interacting tiers, each with its own specific responsibilities. A client tier 102 may include presentation logic, including simple control and user input validation. A mid-tier 104, which may also be known as the application server, provides business processes logic and data access. A data server tier 106 provides business data.

[0003] Multi-tier business rule processing currently requires, to evaluate each rule, large amounts of data to be fetched from the data server 106 and fed into a rules engine of the mid-tier 104. The moving of large amounts of data from one tier to another may increase time and processing required to evaluate a business rule set. Additionally, moving data between tiers may bottleneck the network layer and demand a large memory at the middle tier.

BRIEF SUMMARY OF THE INVENTION

[0004] The present invention relates to a method for preprocessing business rules. The method includes translating at least one business rule into Structured Query Language (SQL), transmitting the SQL to a database tier to act on data of the database tier, and evaluating the data based on the SQL in the database tier

[0005] In another aspect, the present invention relates to a system for preprocessing rules. The system includes a business rules module for storing at least one business rule, a data manager for translating the at least one business rule to Structured Query Language (SQL), and a query data service for transmitting the SQL to a database tier. Evaluation of the SQL against data of the database tier occurs at the database tier.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] For a more complete understanding of the present invention, and for further objects and advantages thereof, reference is made to the following description taken in conjunction with the accompanying drawings in which:

[0007] FIG. 1 is a block diagram of a multi-tiered architecture;

[0008] FIG. 2 is a block diagram of a multi-tiered architecture in accordance with an embodiment of the present invention;

[0009] FIG. 3 is a system diagram illustrating the flow of data in accordance with an embodiment of the present invention;

[0010] FIG. 4 is a class diagram in accordance with an embodiment of the present invention;

[0011] FIG. 5 is a sequence diagram of active event processing in accordance with an embodiment of the present invention; and

[0012] FIG. 6 is a sequence diagram of passive event processing in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0013] Embodiments of the present invention preprocess business rules into Structured Query Language (SQL) and transmit the SQL to a database tier, thereby reducing tier-to-tier movement of data. For example, a user may set up a rule to send correspondence to individuals with a specific zip code. In a traditional business rule implementation, all member records are retrieved from the database tier and moved to the mid-tier for processing a zip code match. Instead, according to embodiments of the present invention, rule matching is executed in the same tier in which data is stored, alleviating the need to move potentially large amounts of data to another tier.

[0014] Referring now to Figure 2, a multi-tiered architecture in accordance with an embodiment of the present invention is illustrated. Services offered by the architecture 200 include business, infrastructure, and data services as shown in the business services tier 202, infrastructure services tier 204, and data services tier 206 respectively. Each business service of the business services tier 202 may be application specific and encapsulate business logic. An infrastructure service, such as those illustrated in the infrastructure services tier 204, may be a general purpose, horizontal service that may not be application specific and does not have business logic. The data services tier 206 relates to services that are general purpose and related to data access. Access to the data may be either read-only or read-write in nature. Although the architecture is illustrated with specific tiers, applications, modules,

databases, etc., it will be understood by one skilled in the art that the multi-tier architecture may be have various different tiers, etc. not illustrated in Figure 2. In addition, the mid-tier, or application server 208, is capable of numerous different deployment configurations and therefore the configuration, as illustrated in Figure 2, is merely exemplary in nature.

[0015] The infrastructure services tier 204 includes a business rules module 210 from which business rules are transferred to a correspondence module 300 to act on the data. The business rules may be set up in a similar manner to database rules, e.g., WHERE class conditions. The business rules may be converted to SQL and sent to the database tier through a Query Data Service 214. The database may then evaluate the business rules against data stored within the database to determine if the business rules are met or not, as described in more detail below. Metadata Service may be used to find Primary Key information and map a Concept/Attribute to a Table/Column.

[0016] Referring now to Figure 3, a block diagram 300 illustrating the flow of data in accordance with an embodiment of the present invention is shown. As previously mentioned, in prior art systems, data is fetched from a database 302 and passed to the business rules service 210 in order for the data to be evaluated (indicated by dashed arrows). The business rules service 210 includes rules that identify a concept (e.g., a table), an attribute (e.g., a column), and a set of values with a relational operator. In addition, multiple rules may be combined together using logical operators. For a given set of data, if the rules are satisfied, then a correspondence letter may be generated using the data and a correspondence template.

[0017] In accordance with an embodiment of the present invention, data is not fetched from the database 302. Instead, rules are fetched from the business rules service 210

(thick arrow) and translated to an SQL query at a Data Manager 306. The Data Manager then passes the query to the database 302. Based on the query, only the matched Primary Key values are fetched to the mid-tier instead of thousands of data records. In the present embodiment, a user may specify correspondence events that trigger correspondence generation. The correspondence events may be either passive or active events. An active event is triggered when a change is made to a table in the database 302. An AuditLog monitor 304 determines when a change has been made to a table and triggers an active event to process as described in detail in Figure 5.

[0018] Passive events are triggered periodically by the scheduling service 212 and do not require a change to a database table. Data flow in accordance with passive event processing is described in detail with reference to Figure 6. Although this diagram is illustrated as executing in a correspondence module, embodiments of the present invention are applicable to various other modules implementing utilizing a rules service. .

[0019] Referring now to Figure 4, a class diagram 400 in accordance with an embodiment of the present invention is illustrated. The class diagram 400 illustrates various objects 306, 402 that are involved in the implementation of an embodiment of the present invention. The relationships between the objects 306, 402 are illustrated via arrows. The class diagram represents the user input to setup a correspondence generation. A Correspondence Schedule 512 associates a Correspondence Type, Correspondence Event, and Correspondence Rule together. The Correspondence Rule has zero or more rule items and each rule item has a left and right hand side of a rule. The left hand side of a rule identifies a concept and attribute pair while the right hand side includes literal values and equates to the right hand side with a relational operator.

[0020] Referring now to Figure 5, a sequence diagram 500 of active event processing in accordance with an embodiment of the present invention is illustrated. Active event processing entails modifying data by a user, and when the data is modified, the data is marked in an AuditLog that triggers the event to process. Involved in evaluating business rules are a Batch Process Audit Log Listener 502, a Correspondence Internal Service 404, Correspondence Manager 506, Audit Manager 508, a Metadata Manager 510, a Correspondence Schedule 512, and the Data Manager 306. The Batch Process Audit Log Listener 502 is triggered, as noted above, to send a batchProcessAuditLog message 516 through the Correspondence Internal Service 504 to the Correspondence Manager 506. The batchProcessAuditLog message 516 triggers a retrieveAuditHeaderMarker message 518 for retrieving the last saved marker from the Audit Manager 508. Data is continued to be retrieved by a getNextAuditHeaderBatch message 520. The getNextAuditHeaderBatch message 520 loops through a batch of, for example, 100 pieces of data at a time. Each AuditHeader is looped through at initiation of a getConceptMetadata message 522 sent from the Correspondence Manager 506 to the Metadata Manager 510. Each AuditHeader lists a table name and, based on the listed table name, the table metadata is retrieved from the Metadata Manager 510.

[0021] The Correspondence Manager 506 retrieves, via a getCorrespondenceEventList message 524, a matching correspondence event list based on given concept information and concept activity (e.g., insert, delete, update, etc.). An isAttributeModified message 526 is transmitted from the Correspondence Manager 506 to the Audit Manager 508 to determine if, in the case of an update activity, an attribute is modified. A getCorrespondenceScheduleList message 528, initiated at the Correspondence Manager 506, retrieves the matching Correspondence Schedules 512. A getCorrespondenceRule

message 530 is sent from the Correspondence Manager 506 to each matching Correspondence Schedule 512 to retrieve rules associated with each Correspondence Schedule 512.

[0022] A getAuditHeaderPrimaryKeyValue message 532 is generated by the Correspondence Manager 506 and sent to the Audit Manager 508 in order to retrieve the primary key value for a row in the AuditLog. Once the primary key value has been retrieved, the Correspondence Manager 506 transmits a getMasterPrimaryKeyValueFromSupportKey message 534 to the Data Manager 306. Based on the primary key value of the table in the AuditLog, a master table row primary key value is retrieved. The Correspondence Manager 506 then sends an isRuleMatchedForMasterPrimaryKeyValue message 536 to the Data Manager 306. This message 536 checks to determine if the rules match for the primary key based on the Correspondence Module 300, master table primary key, and the Correspondence Rule. If the rules match, then a correspondence order is created via a createCorrespondenceOrder message 538. When the processing for each AuditLog is completed, the last processed header marker is saved so that the next event processed may start at the last processed header marker. The last processed header marker is saved via a saveAuditHeaderMarker message 540 created by the Correspondence Manager 506.

[0023] Referring now to Figure 6, a sequence diagram of passive event processing in accordance with an embodiment of the present invention is illustrated. Instead of processing events in an active manner, events may be processed at a predetermined scheduled time as noted above. The sequence may be initiated by a processScheduledJob message 608 from a Process Scheduled Job Listener 602 that is invoked at predetermined intervals by the Scheduling Service 212. The processScheduledJob message 608 delegates

any job waiting for execution to the Correspondence Internal Service 504. The Correspondence Internal Service 504 creates a processScheduledJob message 610 that is, in turn, sent to the Correspondence Manager 506. The processScheduledJob message 610 finds jobs waiting to be processed and calls each job one at a time in a loop. The Correspondence Manager 506 creates a processJob message 612 to initiate processing of a waiting job. A getCorrespondenceJob message 614 aggregates multiple correspondence schedules to be executed as a background job. Each Correspondence Schedule 512 is determined for the waiting job via a getCorrespondenceSchedules message 616 sent to a Correspondence Job 606. The Correspondence Manager 506 obtains correspondence module and correspondence rule information from each Correspondence Schedule 512 via a getCorrespondenceModule message 618 and a getCorrespondenceRule message 620. The Correspondence Manager 506 creates a getRuleMatchedMasterPriaryKeyValue message 622, which is sent to the Data Manager 306, to initiate conversion of the rules to dynamic SQL. The matching primary keys for the correspondence module are retrieved also. The Correspondence Manager 506 creates a createCorrespondenceOrder message 624 for each matched primary key value.

[0024] Although the above embodiments have been illustrated with reference to generating a correspondence order, it will be understood that various modifications maybe made to implement embodiments of the present invention with any business rule that relies on data in a data server for rule evaluation..

[0025] It is thus believed that the operation and construction of embodiments of the present invention will be apparent from the foregoing description. While the method and apparatus shown or described have been characterized as being preferred, it will be obvious

that various changes and modifications may be made therein without departing from the spirit and scope of the invention.